ISSRE99 Panel Proposal

Practical Issues in Implementing Software Reliability Measurement

Allen P. Nikora, panel chair
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA

Norman F. Schneidewind
Division of Computer and Information Sciences and Operations
Naval Postgraduate School
Monterey, CA

John C. Munson
Computer Science Department
University of Idaho
Moscow, ID

John D. Musa
Software Reliability Engineering and Testing Courses

William W. Everett
SPRE, Inc.
Albuquerque, NM

Mladen A. Vouk
Department of Computer Science
College of Engineering
North Carolina State University
Raleigh, NC

Many ways of estimating software systems' reliability, or reliability-related quantities, have been developed over the past several years. Of particular interest are methods that can be used to estimate a software system's fault content prior to test, or to discriminate between components that are fault-prone and those that are not. The results of these methods can be used to:

- More accurately focus scarce fault identification resources on those portions of a software system most in need of it.
- Estimate and forecast the risk of exposure to residual faults in a software system during operation, and develop risk and safety criteria to guide the release of a software system to fielded use.
- Estimate the efficiency of test suites in detecting residual faults.
- Estimate the stability of the software maintenance process.

There are practical issues associated with implementing these techniques in production development environments. For example, one recently-developed technique of estimating a software system's residual fault content relies on measuring its structural evolution throughout its development, and relating the measured structural change to the rate at which faults are inserted. Three issues must be resolved to make use of this technique:

- A baseline for measuring the system's evolution must be established. During implementation, this can be done in quite a straightforward manner, since source code is often placed under the control of a revision control system such as Clear Case or CCC Harvest. It is then a simple matter to define a baseline, and take structural measurements of each piece of source code checked in to the controlled repository with respect to that baseline. However, in the earlier development phases, specification and design artifacts are often much less rigorously controlled - it is frequently not possible to establish a baseline and identify with any certainty subsequent changes to that baseline.

- Structural measurements must be taken of the software system's components. This is quite straightforward with source code – measurements can be taken automatically as part of checking new or modified software components into the repository. This is often more difficult with specification and design artifacts. Frequently, specifications and design are not in a machine-readable form. Furthermore, they are often created using notations with incomplete or ambiguous syntax and semantics. Finally, a system may include specification and design artifacts of more than one type, and these may be incompatible. A standardized notation for representing specification and design artifacts would help to resolve the last two issues – one candidate for such a notation is the Unified Modeling Language (UML).
- In identifying relationships between measured structural change and the fault insertion rate, the number of faults discovered and repaired must be counted. This means that a set of rules for identifying and counting faults must be developed.

In this panel discussion, we discuss these and other practical issues to be addressed in implementing software reliability measurement techniques in a production development environment. The panelists will describe their experiences in implementing measurement programs, the intended and actual results, and suggestions for making them more effective.

Biographies

Allen P. Nikora is a senior member of the Information Systems and Computer Science staff in the Autonomy and Control Section at JPL. He holds a Ph.D. in Computer Science from the University of Southern California. He was principal investigator on a recently-completed task, sponsored by the NASA IV&V Facility, concerned with IV&V issues in achieving high reliability and safety in critical control software. He is currently leading a follow-on task, also sponsored by the NASA IV&V Facility, to address the practical issues of implementing the measurement and risk assessment techniques developed in the previous task on real development efforts. He is also the lead developer of the software reliability modeling tool CASRE, the development of which was funded by the U.S. Air Force Operational Test and Evaluation Center (AFOTEC), and is currently working on functionality enhancements. He has presented papers on software reliability measurement at the International Symposium on Software Reliability Engineering, the International Symposium on Fault-Tolerant Computing (FTCS), and the International Society of Science and Applied Technologies (ISSAT) Conference on Reliability and Quality in Design. He was a contributor to the McGraw-Hill Handbook of Software Reliability Engineering. His research interests are software reliability engineering, software product and process measurement, and software fault tolerance. He is a member of the IEEE Computer and Reliability Societies.

Norman F. Schneidewind is Professor of Information Sciences and Director of the Software Metrics Research Center at the Naval Postgraduate School. He is the developer of the Schneidewind software reliability model which is used by NASA to assist in the prediction of software reliability of the Space Shuttle, by the Naval Surface Warfare Center for Trident software reliability prediction, and by the Marine Corps Tactical Systems Support Activity for distributed system software reliability assessment and prediction. This model is one of the models recommended by the American National Standards Institute and the American Institute of Aeronautics and Astronautics Recommended Practice for Software Reliability. He has published widely in the fields of software reliability and metrics.

Dr. Schneidewind is a Fellow of the IEEE, elected for "contributions to software measurement models in reliability and metrics, and for leadership in advancing the field of software maintenance". The Naval Postgraduate School awarded him a certificate for outstanding research achievements in 1992. He was Chairman of the Working Group that produced the IEEE Standard 1061-1992, Standard for a Software Quality Metrics Methodology and its revision in 1998. In 1993 he was given the IEEE Computer Society's Outstanding Contribution Award "for work leading to the establishment of IEEE Standard 1061-1992". In addition, he was given the IEEE Computer Society Meritorious Service Award "for his long-term committed work in advancing the cause of software engineering standards". He is a member of the IEEE\CS Fellows Committee. He was the General Chair of ISSRE '94.

John C. Munson is a Professor of Computer Science at the University of Idaho. He has been actively engaged in research and publication in the areas of software reliability engineering, software measurement, computer security, and software maintenance. He is a member of the Association for Computing Machinery, the IEEE, the IEEE Computer Society and the IEEE Reliability Society. He has been closely associated with the IEEE International Symposium on Software Reliability and the IEEE International Conference on Software Maintenance serving as a member of the program committee and also as program chair for these conferences. He is currently funded for reliability research by the Jet Propulsion Laboratory, for research in software testing by Storage Technology Corporation and for research in software security by the National Science Foundation.

John D. Musa is one of the creators of the field of software reliability engineering and is widely recognized as the leader in reducing it to practice. He is currently a senior international consultant with a wide variety of clients. He has more than 30 years diversified practical experience as software practitioner and manager. He was formerly Technical Manager of Software Reliability Engineering (SRE) at AT&T Bell Laboratories, Murray Hill, NJ. Musa is a Fellow of the IEEE, and an international leader in software engineering listed in Who's Who in America and American Men and Women of Science. He has published some 100 papers and given more than 175 major presentations. He is principal author of the widely-acclaimed pioneering book Software Reliability: Measurement, Prediction, Application and author of the new book Software Reliability Engineering: More Reliable Software, Faster Development and Testing.

Bill Everett, recently retired from AT&T Bell Laboratories, is currently a consultant and owner of SPRE, Inc., a company providing consulting services in Software Reliability Engineering (SRE).Bill has been involved in SRE for 11 of his 29 years in software development. He is active in professional activities centering on sofware reliability. He is the current chair of the IEEE Committee on SRE. He was the General Chair of ISSRE'97 (International Symposium on Software Reliability Engineering), Finance Chair of ISSRE'96 & 98, secretary of the IEEE CS Press Activities Board, a member of the Program Committees for Testing'96, ASSET'98, ASSET'99, ISSRE'98 and ISSRE'99, a participant in the IEEE Distinguished Visitor's Program, and a past Associate Editor-in-Chief of IEEE SOFTWARE Magazine. He is a chapter contributor to the recently released "Software Reliability Engineering Handbook", a coauthor of the "AT&T Reliability by Design Handbook" and the "AT&T SRE Best Current Practice". He has contributed over two dozen papers, talks, articles, and book chapters on software reliability.

As a technical consultant, Bill works with companies and organizations in applying software reliability. While at AT&T Bell Laboratories, he has held various positions including distinguished member of technical staff, technical manager, technical consultant in software development organizations. He was the team leader in defining a practice for Software Reliability Engineering within AT&T. He has consulted with organizations throughout AT&T on both software reliability and software process engineering.

Bill is a senior member of IEEE, and members of SIAM and ICCA. He received a PhD in Applied Mathematics from the California Institute of Technology and an Engineer's Degree from the Colorado School of Mines.

Mladen A. Vouk received his Ph.D. from the King's College, University of London, U.K. He is currently a Professor of Computer Science at the N.C. State University, Raleigh, N.C., U.S.A. Dr. Vouk has extensive experience in both commercial software production and academic computing. He is the author, or co-author, of over 140 publications. His research and development interests include software engineering (software process and risk management, software testing and reliability), scientific computing (development of numerical and scientific software-based systems, parallel computing, support for scientific problem-solving workflows), computer-based education (network-based education, distance learning, education workflows), and high-speed networks (end-user quality of service, forward error correction in high-speed networks, empirical evaluation of ATM-based high-speed networking solutions). Dr. Vouk is a member, former chairman, and former secretary of the IFIP Working Group 2.5 on Numerical Software. He is also a senior member of IEEE, and a member of IEEE Reliability, Communications, Computer and Education Societies, and of IEEE TC on Software Engineering, ACM, ASQC, and Sigma Xi. He is an associate editor of IEEE Transactions on Reliability, editor of the IEEE TCSE Software Reliability Engineering Newsletter, member of the Editorial Board for the Journal of Computing and Information Technology, and a member of the Editorial Board for the Journal of Parallel and Distributed Computing Practices.